



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/737,639	12/13/2000	Paul F. Austin	5150-50800	2283

7590 01/14/2004

Jeffrey C. Hood
Conley, Rose, & Tayon, P.C.
P.O. Box 398
Austin, TX 78767

EXAMINER

BASOM, BLAINE T

ART UNIT	PAPER NUMBER
----------	--------------

2173

DATE MAILED: 01/14/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/737,639

Applicant(s)

AUSTIN, PAUL F.

Examiner

Blaine Basom

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 20 October 2003.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 43-60 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 43-60 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892) 4) ☐ Interview Summary (PTO-413) Paper No(s). _____
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) 5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ 6) ☐ Other: _____

DETAILED ACTION

Response to Arguments

The Examiner acknowledges the Applicant's cancellation of claims 1-42 and the addition of new claims 43-60. Regarding newly-added claim 43, the Applicant argues that "Mastering Microsoft Office 97" by Moseley and Boodey (hereafter referred to as "Office 97"), as described in the previous Office Action, fails to teach or suggest receiving user input specifying a data source, wherein the user input is received to a program development environment during creation of a program. The Examiner agrees with the Applicant in that Office 97 does not explicitly disclose such a feature. Nevertheless, as is shown below, Risberg (U.S. Patent No. 5,339,392) in fact teaches receiving user input specifying a data source, wherein the user input is received to a program development environment during creation of a program.

Further regarding newly added claim 43, the Applicant submits that neither Risberg nor Office 97 teach programmatically selecting a GUI element based on a data type of data provided by the data source. The Examiner respectfully disagrees, and contends that Office 97 in fact teaches such a feature. As described by the Applicant,

*The Office 97 reference involves pasting a Microsoft Excel Worksheet object, which specifies an application type. Although unclear from the Office 97 reference, Applicant presumes that **the knowledge that the link is made to a Microsoft Excel Worksheet object is used to display the data in a table in the Word document in Figure 3.9.** (See page 8 of Applicant's response). (Emphasis added).*

Thus according to the Applicant, Office 97 teaches programmatically selecting a GUI element, in this case a table, wherein the GUI element is selected based on the knowledge that the link is made to a Microsoft Excel Worksheet object. Analogously, the GUI element is selected based

Art Unit: 2173

on the knowledge that the link is made to Microsoft Excel Worksheet object data. Thus the GUI element is selected based on the data type of the data source.

Lastly, the Applicant argues that, since the Risberg patent relates to a development environment for creating a type of user interface program, whereas the Office 97 reference relates to linking different types of business software applications together, it would not have been obvious to combine the teachings of Risberg and Office 97. The Examiner respectfully disagrees with this argument. Risberg and Office 97 hold many similarities. Like the application taught by Risberg, which is used to construct active documents, Office 97 describes Microsoft Word, an application used to construct documents. More specifically, these Word documents created by Microsoft Word may comprise elements, which like the GUI elements taught by Risberg, display information obtained from one of various external sources such as spreadsheets or other documents (see pages 44 and 45 of Office 97). Also like the GUI elements of Risberg, this information displayed by the element in the Microsoft Word document may be linked to the data source such that live data from the source is used. In other words, if the data changes at the source, the data displayed by the element of the Microsoft Word document similarly changes (see pages 44 and 45 of Office 97). Thus due to Microsoft Word's similarities to the application taught by Risberg, the Examiner contends that one of ordinary skill in the art would have in fact been motivated to implement the favorable features of Word into the application of Risberg.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claim 43 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 5,339,392, which is attributed to Risberg et al. (and hereafter referred to as "Risberg"), and also over Microsoft Office 97, as is described by Lonnie E. Moseley and David M. Boodey in the book entitled *Mastering Microsoft Office 97, Professional Edition* (which is hereafter referred to as "Office 97"). In general, Risberg discloses an application to be used for monitoring and managing complex systems having a plurality of frequently varying data values. More specifically, and regarding the claimed invention, this application allows users to create custom graphical user interfaces in which these data values are displayed, and in which changes in these data values are immediately reflected on the display (see column 1, lines 31-41). These data values, which are received from one or more sources over a network, are particularly displayed via one or more GUI elements. It is thus understood that Risberg teaches a method for configuring a GUI element to publish or subscribe to a data target or data source.

As per claim 43, the application disclosed by Risberg is used to construct GUIs, referred to as "active documents," which are for monitoring financial information such as stock prices. Risberg discloses that the data to be monitored is displayed by a plurality of GUI elements, specifically "quotes," "dynamic graphs," "tickers," or "page fragments" (see column 28, lines 30-67). For example, a page fragment displays a section of data obtained from a financial data

Art Unit: 2173

source. The data displayed via a fragment element is updated in real time (see column 28, lines 62-67). It is understood that quotes, dynamic graphs, and tickers similarly display data from one or more financial sources, except in a different format. To create a page fragment on the active document, a user uses a "Page Fragment tool" and drags, with a mouse, a region on the active document where the page fragment is to be positioned. In response, the page fragment is displayed but contains no information (see column 11, line 65 – column 12, line 4). For the page fragment to display information, the user enters a data source, i.e. "service," into a specific dialog box provided to the user, wherein the data service provides financial data which is displayed in the page fragment (see column 12, lines 5-28). As this data displayed by the page fragment is updated in real time (see column 28, lines 62-67), it is understood that the page fragment is thus configured to receive and display data from the specified data source. It is interpreted that quotes, dynamic graphs, and tickers are created and configured by similar means. Lastly, Risberg notes that the data displayed by a quote, dynamic graph, ticker, or page fragment can be published on a network so that it may be used as a bulletin board or by other users linked to the network (see column 3, line 66 – column 4, line 4). Thus regarding claim 1, Risberg teaches displaying a GUI element, such as a page fragment, on a display; receiving user input specifying a data source, wherein the user input is received to a program development environment during creation of a program, specifically an active document; and, in response to receiving this input, programmatically configuring the GUI element to receive and display data from the specified data source. However, and with respect to the claimed invention, Risberg teaches displaying the GUI element prior to receiving user input specifying at least one of a data source or target. Consequently, Risberg does not teach programmatically selecting a GUI

Art Unit: 2173

element after receiving the user input, wherein the GUI element is selected based on a data type of the data source, and then displaying the selected GUI element in the active document, as is expressed in claim 43.

Similar to the application taught by Risberg, which is used to construct active documents, i.e. GUIs, Microsoft Word is an application used to construct documents. More specifically, these Word documents created by Microsoft word may comprise elements, which like the page fragments taught by Risberg and described above, display information obtained from one of various external sources such as spreadsheets or other documents (see pages 44 and 45 of Office 97). Also like the page fragment of Risberg, this information displayed by the element in the Microsoft Word document may be linked to the data source such that live data from the source is used. In other words, if the data changes at the source, the data displayed by the element of the Microsoft Word document similarly changes (see pages 44 and 45 of Office 97). With respect to the claimed invention, a user specifies a data source by selecting the data source with a mouse cursor and then selecting a “paste” option in the “edit” menu of Microsoft Word. In response, the selected data source information is displayed in an appropriate GUI element in the Microsoft Word document. For example, when pasting a spreadsheet source, the data appears in the Microsoft Word document via a table-like GUI element (see figure 3.9 on page 46 of Office 97). Moreover, and like the teachings of Risberg, it is understood that the user may also specify a particular data source by selecting the source in an “Insert File” dialog box provided to the user, as apposed to copying and pasting the source. The selected source data is displayed via an appropriate GUI element in the Word document in response (see pages 380 and 381 of Office 97). Thus Office 97 presents a technique for configuring a GUI element to subscribe to a data

Art Unit: 2173

source, wherein the user specifies a data source either by copying and pasting the particular data source, or by using a dialog box. In response, a GUI element is displayed which receives and displays data from this data source, as is shown above. As further taught by Office 97, this particular GUI element is based on the data source. For example, figure 3.9 on page 46 shows a GUI element, which is displayed in response to the specification of a spreadsheet data source. As shown in figure 3.9, this GUI element is a table. Figure 14.2 on page 382 on the other hand shows a GUI element, which is displayed in response to the selection of a document data source. As shown in figure 14.2, this GUI element simply comprises the text of the source document. Thus the GUI element is different based on the data source; a spreadsheet data source is displayed via a table-like GUI element, while a document data source is displayed via textual GUI element. Consequently, it is understood that Office 97 teaches programmatically selecting a GUI element based on the data type of the data source, and displaying this GUI element in response to the user input specifying the data source.

It would therefore have been obvious to one of ordinary skill in the art, having the teachings of Risberg and Office 97 before him at the time the invention was made, to modify the application taught by Risberg, such that instead of creating a GUI element and then selecting a data source with which to associate the GUI element, the data source is first selected and then in response, a GUI element associated with the data source is automatically selected and displayed, as is done in Office 97. It would have been advantageous to one of ordinary skill to utilize such a combination because a more efficient means for configuring the active document to display data from a data source results; instead of having to create a GUI element and then select a data source with which to populate the GUI element, as is done by Risberg, with the combination of

Art Unit: 2173

Risberg and Office 97, a user simply has to select a data source – the GUI element is automatically created in response. Thus this method taught by Risberg and Office 97 is implemented as a software program (for example, see column 26, lines 53-68 of Risberg). Consequently, it is understood that it is executed by a computer system, the computer system having: display device to present the active document to the user; a processor to execute the software program; and a memory coupled to the processor to store the program. Such a memory is considered a memory medium like that of claim 43.

Regarding claim 44, the above-described combination of Risberg and Office 97 teaches programmatically selecting and displaying a GUI element in response to specifying a data source. As shown above, the user specifies a data source either by highlighting the particular data source, or by using a dialog box. In response to selecting and displaying the GUI element, the GUI element automatically receives and displays data from the data source, as is further shown above. Thus the GUI element is automatically configured to receive and display data without user programming and without the user input specifying source code.

Regarding claim 47, the combination of Risberg and Office 97 teach a method whereby, as is shown above, a user specifies a data source either by copying and pasting the particular data source, or by using a dialog box. In response, a GUI element is displayed which receives and displays data from this data source, as is shown above. As further taught by Office 97, this particular GUI element is based on the data source. For example, figure 3.9 on page 46 shows a GUI element, which is displayed in response to the specification of a spreadsheet data source. As shown in figure 3.9, this GUI element is a table. Figure 14.2 on page 382 on the other hand shows a GUI element, which is displayed in response to the selection of a document data source.

Art Unit: 2173

As shown in figure 14.2, this GUI element simply comprises the text of the source document.

Thus the GUI element is different based on the data source; a spreadsheet data source is

displayed via a table-like GUI element, while a document data source is displayed via textual

GUI element. Office 97 further teaches that the particular GUI element is based on the data from

the data source. For example, the GUI element in figure 3.9, which as described above is

displayed in response to the selection of a spreadsheet data source, is a table comprising 1

column and 3 rows. Figure 14.8 on page 387 also shows a GUI element which is displayed in

response to the selection of a document data source. As shown in figure 14.8, this GUI element

is a table comprising 4 columns and 12 rows. Comparing figure 14.6 on page 386 and figure 3.7

on page 44, which respectively show the spreadsheet data sources for the above described GUI

elements, it is noted that the source which is displayed in the larger GUI element, i.e. the GUI

element of figure 14.8, is the spreadsheet comprising more data rows and columns (that of figure

14.6). Thus the GUI element is different based on the data of the data source; a spreadsheet

source with many columns and rows of data is displayed in a larger GUI element than a

spreadsheet source with fewer columns and rows of data. Consequently, it is understood that the

combination of Risberg and Office 97 described above teaches receiving data from the data

source, and programmatically analyzing the received data to programmatically determine a GUI

element operable to display the received data.

As for claim 48, the combination of Risberg and Office 97 teach a method whereby, as is shown above, an application is used to display and configure a GUI element to receive and

display data from a specific data source. Such a GUI element is displayed and configured in

response to specifying a data source. The data received and displayed from the data source is in

Art Unit: 2173

one of many different possible formats. For example, the data may be spreadsheet data or text data, among others, as is described above. In any case, since the data received from the data source may be in one of many different formats, which must be analyzed in order to ascertain how to display the data, it is understood that the data must be in a self-describing format. In other words, it is understood that the data itself at least partially describes how it is to be displayed. Consequently, the combination of Risberg and Office 97 described above teaches that the data received is in a self-describing format, and wherein a GUI element is automatically determined that is operable to indicate this data.

With respect to claim 49, the application disclosed by Risberg, which as described above is used to construct GUIs for viewing financial information, is implemented on a first computer which accesses over a network a second computer, specifically a server (see column 2, lines 39-48). The application receives data from the server and displays the data via one or more graphical elements (see column 2, lines 27-48). Thus it is understood that the method taught by the combination of Risberg and Microsoft Office 97, as is described above, is executed on a first computer, whereby this first computer is operable to connect to a second computer over a network. It is further understood that this second computer comprises a data source, whereby the GUI elements taught by Risberg and Office 97 may be configured to connect to the second computer and receive and display data from this data source.

Referring to claim 50, the application disclosed by Risberg, as modified by the teachings of Microsoft Office 97, is used to construct GUIs, or more specifically "active documents," as is described above. Risberg further discloses that each active document may comprise one or more scripts which define the functionality of various components of the active document (see column

Art Unit: 2173

2, lines 27-39). In any event, an active document is considered a computer program, as it is executed on a computer to access data over a network and display it on the computer. Thus the GUI elements taught by Risberg are associated with a first computer program, specifically an active document. These GUI elements are included in a user interface associated with an active document (for example, see figure 1 of Risberg, which shows the user interface of an active document; a quote, which is designated by reference number 18; a ticker, which is designated by reference number 20; a dynamic graph, which is designated by reference number 22; and a page fragment, which is designated by reference number 24, are displayed in this user interface).

As per claim 52, Microsoft Office 97 teaches that the data source, with which a GUI element is linked, may be a file. For example, the data source may be a document or spreadsheet file, as is described above. Thus it is understood that with the method taught by the combination of Risberg and Microsoft Office 97, as is described above, the data source or data target is one from the group consisting of an HTTP server, an FTP server, and OPC server, an SNMP server, a DataSocket server, and a file. More specifically, the data source is a file.

Regarding claims 55 and 56, the data received and displayed by the GUI elements disclosed by Risberg is displayed in real time (for example, see column 2, lines 39-48). Consequently, the data is considered live data. Moreover, the data is financial data, which is obtained or measured by one of a plurality of services, such as "MarketFeed 2," "Telerate," or "Reuters" (for example, see column 28, lines 40-47). The data is therefore also considered measurement data. Therefore, the combination of Risberg and Office 97 described above teaches providing user input specifying a data source with which to associate a GUI element, and

Art Unit: 2173

in response, automatically displaying and configuring the GUI element to receive and display live data and measurement data from this specified source.

Regarding claims 57 and 59, the above-described application of Risberg and Office 97 is implemented as a software program (for example, see column 26, lines 53-68 of Risberg). Consequently, it is understood that it is executed by a computer system, the computer system having: display device to present the active document to the user; a processor to execute the software program; and a memory coupled to the processor to store the program. Such a memory is considered a memory medium like that of claim 57. Similarly, such a computer system implementing the application is considered a system, like that recited in claim 59.

Claims 45, 46, 53, 54, and 60 are rejected under 35 U.S.C. 103(a) as being obvious over the combination of Risberg and Office 97, which is described above, and also over U.S. Patent No. 5,959,621, which is attributed to Nawaz et al. (and hereafter referred to as "Nawaz"). Specifically referring to claim 45, the combination of Risberg and Office 97 teaches a method like that of claim 43, wherein a user may specify a data source via a dialog box, as is shown above. In response, a GUI element is automatically selected and configured to receive and display data from this data source, as is shown above. Risberg further discloses that the data source may be a server over a network (see column 2, lines 39-48). The combination however does not explicitly disclose that this input specifying a data source is a URL, as is expressed in claim 45.

Like the teachings of Risberg and Office 97, Nawaz presents a method for configuring a GUI element, specifically a ticker, to receive and display data from a specific data source. With

Art Unit: 2173

further similarity to the teachings of Risberg, this data may comprise financial data, namely stock prices (see column 3, lines 27-29). Regarding the claimed invention, Nawaz teaches that the data is received and displayed in the ticker from sources specified by URLs (see column 12, lines 23-40).

It would have therefore been obvious to one of ordinary skill in the art, having the teachings of Risberg, Microsoft Office 97, and Nawaz before him at the time the invention was made, to modify the method taught by the combination Risberg and Office 97 such that the data sources are specified by URLs, as is done by Nawaz. One would have been motivated to create such a combination because, as is demonstrated by Nawaz, URLs provide well-known and commonly used identification means for identifying data on a network.

With regard to claim 46, the combination of Risberg, Microsoft Office 97, and Nawaz described above teaches receiving user input specifying at least one data source or target, wherein particularly, this user input specifies a URL of the data source or target. As shown above in the rejection for claims 47 and 48, Risberg and Office 97 further teaches automatically determining an appropriate GUI element to display in response to such input, wherein specifically, the GUI element is determined based on the specified data source. For example, as is shown above, a document data source, which has a "doc" file extension, is displayed in a different type of GUI element than a spreadsheet data source, which has a different file extension. Consequently, it is interpreted that the GUI element is determined based on the file extension of the specified data source. As is known in the art, such file extensions are included in the URL specifying the location of such files. It is therefore understood that the combination

Art Unit: 2173

of Risberg, Office 97, and Nawaz described above further teaches automatically determining an appropriate GUI element to display based on a file extension specified by the URL.

Specifically regarding claims 53 and 54, the combination of Risberg and Office 97 teaches a method for configuring a GUI element, wherein as shown above in the rejection for claim 43, the method comprises receiving user input specifying a data source, and then displaying an appropriate GUI element with which to associate the data source. In response, the GUI element is automatically configured to receive and display data from the specified data source. Risberg further teaches that the data displayed in a constructed active document, which is interpreted to be the data displayed via such a GUI element, may be published on a network (see column 3, line 66 – column 6, line 4). Thus it is understood that Risberg discloses configuring a GUI element to publish data. However, and with respect to the claimed invention, neither Risberg nor Office 97 specifically teaches the steps undertaken to publish such data to a specific target. In other words, the combination of Risberg and Office 97 does not explicitly teach receiving user input specifying a first data target with which to associate the GUI element, and programmatically configuring the GUI element to publish data associated with the GUI element to this first data target, as is expressed in claim 53. Consequently, the combination also does not disclose inputting a data source and a data target, wherein the source is the same as the target, as is recited in claim 54.

Like the combination of Risberg and Office 97, Nawaz presents a method for configuring a GUI element, specifically a ticker, to receive and display data from a specific data source. With further similarity to the teachings of Risberg, this data may comprise financial data, namely stock prices (see column 3, lines 27-29). With respect to the claimed invention, the ticker

Art Unit: 2173

disclosed by Nawaz, like the GUI elements of Risberg, may be configured to publish data associated with the ticker to one or more specific targets. Specifically, a user uses a "posting page" associated with the ticker to determine which users, i.e. targets, receive data associated with the ticker (see column 11, lines 40-55).

It would have therefore been obvious to one of ordinary skill in the art, having the teachings of Risberg, Office 97, and Nawaz before him at the time the invention was made, to modify the method taught by the combination of Risberg and Office 97 such that a page similar to the posting page of Nawaz is utilized to display and configure a GUI element to publish data to a specific data target. In other words, it would have been obvious to modify the method of Risberg and Office 97 such that, with a posting page, a user specifies a first data target with which to associate the GUI element, and consequently, the GUI element is automatically displayed and configured to publish data associated with the GUI element to this data target. One would have been motivated to create such a combination because, as is demonstrated by Nawaz, a posting page allows a user to efficiently delineate who should and should not receive data associated with a GUI element. The provision of such a utility is beneficial when publishing data, as is shown by Nawaz. It is thus understood that, with this combination of Risberg, Office 97, and Nawaz, a user may specify both a data source and a data target with which to associate a GUI element. Moreover, it is understood that the specified data source may be the same as the specified data target. In such a case, the GUI element would be automatically configured to receive and display data from the remote data source, and publish data to the specified data target.

As per claim 60, the above-described application of Risberg, Office 97, and Nawaz is considered to teach a method like that recited in claim 60, which is for configuring a graphical user interface element to publish and subscribe to data

Claims 51 and 58 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Risberg and Office 97, which is described above, and also over U.S. Patent No. 5,291,587, which is attributed to Kodosky et al. (and hereafter referred to as "Kodosky"). As shown above, the combination of Risberg and Office 97 presents an active document containing one or more GUI elements, which are automatically configured to receive and display data from user-specified data sources. Since an active document contains GUI elements, it is considered a graphical program. Moreover, Risberg discloses that a user may create one or more scripts which define the functionality of the active document, whereby the scripts access this data provided by the GUI elements and thus function according to this data. For example, Risberg discloses that a script may be created, which is executed if the data received by a quote element is in a pre-defined "normal" state. On the other hand, if the data received by a quote element is in a pre-defined "alert" state, an alternative script is executed (see column 10, lines 4-27). These scripts may perform one or more operations, including selecting user-specified objects and editing these objects (see column 16, lines 62-68). However, and with respect to the claimed invention, Risberg does not explicitly disclose that the active document includes a block diagram comprising a plurality of connected nodes, wherein as recited in claim 51, the connected nodes visually represent functionality of the active document.

Art Unit: 2173

Like Risberg, Kodosky presents an application that is used to construct graphical user interfaces, referred to as a "front panels," which provide for monitoring and managing complex systems having a plurality of frequently varying data values. Such complex systems are namely data collecting instruments (see column 2, lines 6-18 and lines 44-52). Regarding the claimed invention, Kodosky teaches that a "graphical editor" may be used to construct a graphical diagram to specify the functionality of the interface (see column 7, lines 36-40). Figure 22 shows an example of such a functional diagram. As shown in figure 22, the functional diagram is essentially a block diagram comprising a plurality of connected nodes, each node representing a particular function or element of the front panel. As further taught by Kodosky, when a GUI element is placed in a front panel, a node representative of the GUI element is automatically placed in the functional diagram (see column 14, lines 59-64).

It would therefore have been obvious to one of ordinary skill in the art, having the teachings of Risberg, Office 97, and Kodosky before him at the time the invention was made, to modify the application taught by Risberg and Kodosky such that, like the application taught by Kodosky, it includes a block diagram defining the functionality of the active document created by the application. It would have been advantageous to one of ordinary skill to utilize such combination because an easier means for defining the functionality of the active document would have been obtained, as is taught by Kodosky (see column 2, lines 53-65, and column 3, lines 30-35).

Regarding claim 58, the above-described application of Risberg, Office 97, and Kodosky is implemented as a software program (for example, see column 26, lines 53-68 of Risberg). Consequently, it is understood that it is executed by a computer system, the computer system

Art Unit: 2173

having: display device to present the active document to the user; a processor to execute the software program; and a memory coupled to the processor to store the program. Such a memory is considered a memory medium like that of claim 58.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

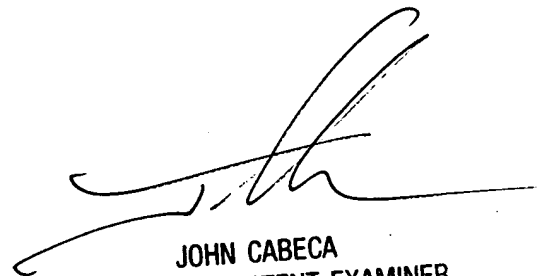
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Blaine Basom whose telephone number is (703) 305-7694. The examiner can normally be reached on Monday through Friday, from 8:30 am to 5:30 pm.

Art Unit: 2173

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (703) 308-3116. The fax phone number for the organization where this application or proceeding is assigned is (703) 746-7238.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 305-3900.

btb



JOHN CABECA
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100